NEWS
OF THENATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
PHYSICO-MATHEMATICAL SERIES
ISSN 1991-346X
Volume 2, Number 336 (2021), 48 – 55 **https://doi.org/10.32014/2021.2518-1726.20**

UDC 004.9
IRSTI 28.23.39

A. Satybaldiyeva[1], A. Ismailova[1], R. Moldasheva[1], A.Mukhanova[2], K. Kadirkulov[1]

[1]Saken Seifullin Kazakh Agrotechnical University, Nur-Sultan, Kazakhstan;
[2]L.N.Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan.
E-mail: satekbayeva@gmail.com

ABSTRACT DATA TYPES FOR KNOWLEDGE REPRESENTATION AND SPECIFICATION OF MULTI-AGENT SYSTEMS

Abstract. Distributed system is a group of decentralized interacting executers. Distributed algorithm is the communication protocol for a distributed system that transforms the group into a team to solve some task. Multiagent system is a distributed system that consists of autonomous reactive agents, i.e. executers which internal states can be characterized in terms Believes (B), Desires (D), and Intentions (I). Multiagent algorithm is a distributed algorithm for a multiagent system.

The article discusses the basic concepts of agents and multi-agent systems. Also, two problems of multi-agent algorithms for representing knowledge in the context of Social Software Engineering are considered. A number of new multi-agent algorithms are presented, and their correctness is proved. The main characteristics of agents are provided, such as autonomy, proactivity, social ability, and reactivity; also, agents can have such additional characteristics as persistence, reasonability, performance, mobility, personality, and rationality. A number of new multi-agent algorithms are presented, and their correctness is proved. Two statements have been proved for solving RAM and MRP problems. This time we address a social issue of agent anonymity and privacy in these algo-rithms.

Keywords: multiagent systems and algorithms, assignment problem, safety and progress properties, algorithm verification, privacy, anonymity, Social Software.

Overview of multi-agent systems. Users of computer software are becoming more and more dependent on computer systems. One of the many reasons for this is that computer systems have the ability to disseminate information through efficient interaction within mobile and (or) physical networks [1]. This capability has made computer systems attractive because they can solve problems or needs of various users and organizations. Computer systems also have the ability to reflect organizational capabilities and priorities. This includes the ability to represent the interests and objectives of various users (for example, the best auction price, the most relevant search results, etc.). However, reliance on computer systems and the dynamic nature of computing environments (e. g., the Internet) have caused increased requirements (e. g., reliability, performance, and scalability) to computer systems. This has generated interest towards research focused on automated computer systems such as single agents and multi-agent systems [2, 3].

Advances in artificial intelligence research in the mid-1960s and early 1970s led to the development of agents in the 1970s. In 1966, Professor Joseph Weizenbaum of Massachusetts Institute of Technology wrote the first software agent known as ELIZA [2, p. 550].

Research on software agents in the 1970s and 1980s was very limited, being an additional research and did not focus on language analysis, knowledge representation, nor on automatic and machine learning until the advent of the World Wide Web in the early 1990s. With the advent of the World Wide Web, a multi-agent system was created – the Worm search engine. The purpose of that agent was to track web pages [4]. Other agents were created just as computer technologies became more advanced. In 1997, the first sales agent, RoboShopper, was created to help customers or users to shop over the Internet. RoboShopper accomplished that purpose by searching for items online and comparing site item prices.

Microsoft Office Assistant also released an “office assistant”, which consisted of agent programs designed to interact and assist the user based on the Microsoft Office Assistant [5]. The sites were designed to search for recently hosted web pages and add the found pages to the major search engines.

It is not hard to see that such well-known computer programs as the Google search engine and Microsoft Office Assistant are agents or exhibit some form of agent behaviour. They are used frequently, yield satisfactory results to the user, and perform other additional functions.

Since the early 1980s, there has been an increase in research on multi-agent systems. This is because most real-world hard problems involve distributed open systems. An open system is flexible and scalable [4, p. 24].

Agent Definitions. Due to the variety of research areas and examples, there is no standard definition of agents. Below are some examples of such definitions:

SodaBot Definition – “Software agents are programs that participate in dialogues and negotiations, and coordinate the transfer of information” [5, p. 1495]. Sodabot is a development environment for software agents curated by Michael Cohen at Massachusetts Institute of Technology. By his definition, agents communicate with one another about information by perceiving and acting on the environment. This definition excludes most standard programs as agents.

Definition by Pattie Maes – “Autonomous agents are computing systems that represent a defined complex dynamic environment, act autonomously in that environment and thus implement a set of goals or objectives for which they are designed.” According to Mays’ definition, agents are autonomous, that is, they can act without targeted training. They also define the environment in which agents act as strictly complex and dynamic. This means that the conditions in which agents operate constantly change and require to adapt to changes in the environment [6, p.110].

Hayes-Roth Definition – “Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; solution of problems; and drawing conclusions and determining actions” [7]. According to Hayes-Roth’s definition, INTELIGENT agents must reason before they act. This means that reflex agents are not INTELIGENT agents because they react spontaneously, for no reason.

M. Wooldridge’s definition – “Agents are simply computer systems that are capable of autonomous actions in some environment in order to satisfy their output. As defined by Wooldridge, agents act on their own behalf. Agents can also coordinate with one another to solve problems or satisfy goals without human intervention [8].”

More definitions can be found in the work done by Honavar. From the above definitions, it is clear that agents are autonomous, have the ability to communicate with other agents or users, and are part of the environment. The above definitions can also be used to obtain other important characteristics of the agent [9].

Single (or individual) agents and multi-agent systems are the research part in the field of artificial intelligence and distributed systems, respectively. Research in terms of individual agents on the structure and internal behaviour of an individual agent, which involves autonomy, mobility, learning, and algorithm (-s) is applied to solve problems, etc. Research within the framework of multi-agent systems is on the coordination of several agents and is used to solve problems or perform tasks, i. e. external behaviour of agents. Coordination of multiple agents includes defining interaction protocols, intermediaries, communication languages, etc. Individual agents and multi-agent systems are designed to provide or achieve the most basic qualities of agency systems (agency, intelligence, and mobility) operating in various environments.

There are various definitions for the term “agent”, as discussed above. These definitions can be used to identify the characteristics of an agent. Characteristics can help distinguish a multi-agent system from a typical application. Below are the main characteristics of the agent [10]:

- autonomy – the agent can act independently, without direct human intervention and can control its own actions and internal state;
- proactivity – it is a purposeful agent and it is capable of accomplishing goals without asking the user or other agents. It is also able to adapt to changes in the environment;
- social ability – the agent can interact with people or other agents using the agent’s communication language; and

– reactivity – the agent is able to perceive and respond to changes in the environment immediately or within a short period of time.

In addition to the characteristics of the agents described above, the agent can also have the following additional characteristics:

– persistence – the agent continues to operate with the process continuously until it reaches the desired result.

– reasoning – the agent's reasoning about its actions before deciding at the output.

Performance – the agent is capable of achieving the desired results.

Mobility – the agent has the ability to move from one platform to another.

Personality – the agent has the ability to manifest a characteristic of human nature.

Rationality – the agent chooses one or another action that maximizes its own assessment of the activity, taking into account the sequence of perceptions and everything that is built into the agent's knowledge.

Unlike the Turing test, an agent that has the properties defined above is considered to be an autonomous agent, and hence the terms agent and autonomous agent are interchangeable.

The basic concepts of agents and multi-agent systems are discussed in this article. Also, two problems of multi-agent algorithms for representing knowledge in the context of social software engineering are considered. A number of new multi-agent algorithms are presented, and their correctness is proved.

Multi-agent Systems and Algorithms. Many problems that are solved by multi-agent algorithms can be considered examples of problems of Social Software Engineering, i. e. of a relatively new scientific paradigm, the essence of which is as follows. In the modern world, many social requirements and procedures are of a very clearly described algorithmic nature. Therefore, these requirements can be represented in the form of (semi) formal specifications, and procedures can be represented programmatically (in some programming language or semi-formal pseudo code), after which the properties of these procedures can be investigated by methods of program analysis and verification. Well, and the results of formal analysis or verification can be interpreted in socially significant terms. And although they started talking about Social Software Engineering only in the 21st century, the study by H. Steinhaus, B. Knaster, and S. Banach of the problem of slicing the pie [11] can be considered to be the first example of application of this paradigm.

A distributed system is a group of decentralized interacting performers. A distributed algorithm is a protocol for the interaction of performers in a distributed system, which converts a decentralized group into a team that jointly solves a certain problem [11, p. 104].

A multi-agent system is a distributed system consisting of agents. An agent is an autonomous (perceiving the world divided into "itself" and "the environment", which includes everything else) and a reactive (capable of interacting with the environment and responding to environmental influences) object (in the object-oriented sense), the internal state of which can be characterized in terms of "opinions", "perceptions" or "believes" (Believes), objectives (Desires), and intentions (Intentions) of the agent. A multi-agent algorithm is a distributed algorithm for a multi-agent system.

The agent's belief[1] is a set of its opinions about itself and the environment, which may be incomplete, inconsistent, and generally incorrect (not true), whereas the agent's knowledge is its opinions that correspond to reality[2]. The objectives of the agent are its long-term tasks and responsibilities, which can also be inconsistent. The agent's intentions are used for short-term planning. The agent is responsive in the sense that it can change its belief, objective or intention after interacting with other agents or the environment, but each agent is still autonomous, which means that the change in its internal state depends only on itself, and not on the environment ... Agents of the described type are commonly referred to as BDI agents.

A rational agent has a clear idea of its preferences among potential objectives and always chooses the action that has the highest priority and (from its point of view) promises the maximum benefit; there is a distinction between a complete and limited rationality: they differ in the cognitive and deductive abilities of agents. An intelligent agent is an agent controlled not by an imperative or functional deterministic program, but by a non-deterministic logical program represented by behaviour rules of the form, where $C_1, \ldots, C_k$ are logical conditions on local variables and agent communication channels, $C_1 \& \ldots C_k \Longrightarrow$

fork $A_{1,1}; \dots A_{1,m} || \dots A_{n,1}; \dots A_{n,m}$ join is a construction of parallel execution of several branches, and $A_{1,1}; \dots A_{1,m}, \dots A_{n,1}, \dots, A_{n,m}$ are actions from the number of assignments of new values to local variables, reading data from input channels and ascribing them to local variables, and sending local data to output channels.

About RAM Problem Solving

At first glance, RAM and MRP are different problems. First, in RAM, agents are rational, while in MRP, agents are just robots that do not care at all about their benefits (about the length of the traversed route, for example). Further, the MRP problem has an obvious geometric interpretation, but it is not at all obvious that there are many routes without intersections, and, therefore, that it can be constructed by any multi-agent algorithm whatsoever; at the same time, the RAM problem does not have a geometric interpretation, but it is obvious that there is such a choice of sellers by buyers that any exchange of sellers by any pair of buyers cannot reduce the total amount to be paid to all buyers together.

But from the algorithmic point of view, both RAM and MRP problems are closely related, since their solutions belong to the class of so-called wave algorithms. This class of distributed algorithms has the following general properties.

– completion: all agents complete their operation in a finite time;
– decision: each agent has a final decision-making moment; and
– interdependence: the decision of each agent affects all agents.

The individual beliefs of each agent in both algorithms are represented by two NC and CF integer counters:

– the current value of NC (Number of Conflicts) is the upper estimate of the number of agents with which the agent may have a conflict of intentions right now;
– the current CF (Conflict Free) value is the lower estimate of the number of agents that have no conflicts at all.

Paper [12] presented two imperative wave algorithms LSM (Look for Salesman) and SWP (SWaPping). LSM algorithm assumes that every buyer b has a fine* f_b for repeating its bids for any salesmen. The following two propositions about LSM and SWP were proved in [12, p. 113].

Proposition 1.

If *a multiagent system with fair communication consists of* $m > 0$ *buyers each of which would like to make an individual deal with some of* $n \geq m$ *salesmen, it is common knowledge (in the system) that all buyers are agents executing algorithm LSM, and for every buyer b its fine* f_b *is always less than the minimal price in its price-list*,

then *every agent will eventually terminate, it will know upon termination that nobody in the system will never compete for its current salesman cur_sman, and (hence) it will be able to make a deal with this salesman*.

Proposition 2.

If *a multiagent system with fair communication consists of* $m > 0$ *buyers each of which knows some initial individual salesman among* $n \geq m$ *salesmen, it is common knowledge (in the system) that all buyers are agents executing algorithm SWP*,

then *every agent will eventually terminate, it will know upon termination an individual salesman (that may be different from the initial one), and it will know that it is impossible to reduce by swapping the total price all buyers have to pay*.

These two propositions solve RAM problem, but under assumption of fair communication. This assumption constitutes that if anyone of agents ever wants to communicate with any another agent, sooner or later the communication session between them will surely take place. In this paper we will not discuss how to guarantee this fairness, but we would like to point to one option that solves the problem: one can assign priorities to buyers and allow seniors to initiate communication with juniors.

* Time is money.

Another drawback of the algorithms LSM and SWP is their imperative nature. And, therefore, agents, guided by these algorithms are rational, but have not yet become intelligent.

Social Software Engineering and Agents

At the same time RAM is closely related to the classic Cake Cutting Problem (CC-problem), also known as Fair Division Problem [6, p.124] that has been introduced by a group of Polish mathematicians, Hugo Steinhaus, Bronislaw Knaster and Stefan Banach.

The CC-problem is to divide an infinitely dividable resource ("cake") in such a way that all recipients believe that they have received a fair. Special cases of the problem are proportional and envy-free division. A division is said to be envy-free if each recipient believes (at the moment of reception) that according to his measure no other recipient has received more than he has of a heterogeneous cake; in contrast, a proportional division deals with a homogeneous cake where each of m recipients have to receive exactly $1/\mathrm{m}$ of the cake's volume.

To explain connections between RAM- and CC-problems, it is enough to reformulate a RAM as follows:

The cake is cut on $n > 0$ pieces, which should be divided among $m > 0$ recipients. Each recipient inb $\in [1..m]$is the intelligent rational agent to whom exactly one of piece of a cake is necessary, and it knows the scale of value of pieces$\{p(b,s) \geq 0: s \in [1..m]\}$. All recipientscan (in P2P-manner) communicate, negotiate, make concessions, flip (individual change) and swap (pairwise exchange) their pieces of a cake so that all concessions and swaps must be rational for both participating recipients. However, each recipient can buy the chosen piece if and only if he/she knows that nobody else will ever apply for this piece of cake.

Problem: Design a multiagent algorithm for recipients, which will allow each agent sooner or later to get a piece of cake and, besides, guarantees that any pairwise swap of pieces can't reduce total value.

Differences between RAM- and CC-problems are evident: in CC-problem a cake is an infinitely dividable resource, while in RAM-problem a "resource" has been cut already onto "salesmen"; solutions of the CC-problem may be sequential, while solutions (if any) of RAM must be multiagent (i.e. distributed, parallel and concurrent) by the problem statement. But even multiagent solution of CC-problem can be unsuitable for a RAM-problem. For example, the classical envy-free solution of CC-problem for two participants consists in the following: one agent cuts the cake so that any of two pieces will satisfy him/her and the second chooses from these two pieces which satisfies her/him. As it is easy to see if the cake is already cut on two pieces, and both agents wish the same piece, the system of these two agents will get to the deadlock.

But in spite of these differences, RAM- and CC-problems have something in common since they both are examples of a new research paradigm of Social Software [13].

In the modern world very many social requirements and procedures have algorithmic character. These requirements can be written as (semi-)formal specifications and procedures – software (in a pseudocode). Then the properties of these procedures can be analyzed and verified by formal methods. Well, the results of the formal analysis or verification may be interpreted in socially significant terms. And though about Social Software started talking only in a XXI century, but it is possible to consider as the first example of application of this paradigm research of the Cake Cutting Problem by H. Shteinhaus, B.Knaster and S. Banach.

This is exactly how in the paper [12, p. 115], LSM and SWP algorithms were formalized and Proposition 1 and 2 were verified (using the methods of verification of imperative programs). And the conclusion that can be drawn from these results is that conflicts between rational agents over the seller (a piece of the cake and so on) can be resolved through penalties for intransigence, but these penalties must be high enough (see Proposition 1).

Cryptographic Aspect Research

As part of the study of the cryptographic aspect of the RinS problem, we will assume that the input data (coordinates of robots and shelters) are taken from some finite set of points in $\mathbb{R}^k, k \geq 2$ space, all the coordinates of which are numbers represented by a finite number of digits of some (fixed) positional number system. Let us note that in this case the meaning of the shelter distribution protocol is to calculate some function of coordinates of robots and shelters, the value of which is the desired distribution. By virtue of the Oblivious Transfer secure multi-party computation theorem [14, 15, 16] *on confidential calculations*, there is a way to calculate this function, in which the participants will not receive any additional information about one another's input data. Unfortunately, the direct description of this method is too cumbersome.

Therefore, we will somewhat simplify our task: we will assume that the participants are using some kind of clicks-based shelter distribution protocol. Our objective, therefore, is to construct a click protocol in which the participants would not reveal any additional information about their coordinates to one another. We will consider the participants "semi-honest" (honest, but curious). This means that they strictly follow the protocol, although they try to extract as much additional information from one another's messages as possible. This approach is fully consistent with the interpretation of participants as agents who strictly follow their program and, moreover, are interested in the correctness of the protocol.

Cryptographic Aspect: Results

Proposition 3. *Let $S \in \mathbb{R}^k$ be an arbitrary finite set of points, all the coordinates of which are numbers with a fixed number of digits each in some fixed positional number system.*

Then:

– there is a restriction of the simple click protocol to *S* set, in which the agents do not communicate their coordinates to one another;

– there is a restriction of the click protocol with comparisons to *S* set, in which the agents do not communicate to one another their distances to covers.

Proof. Let us take a closer look at the simple click protocol. (The click protocol with comparisons is treated similarly). Note that the intersection of two straight-line routes [R_1, S_1] and [R_2, S_2] is equivalent to the conjunction of the three conditions as follows:

– both segments lie in the same plane;

– points R_1 и S_1 are separated by a straight line *l* (R_2, S_2); and

– points R_2 and S_2 are separated by a straight line *l* (R_1, 1_2).

All these three conditions are easily expressed by means of analytical geometry in the form of equalities and inequalities between (fixed) arithmetic expressions from the coordinates of robots R_i, R_j and covers S_i, S_j. Due to the finiteness of S set and the finiteness of the representation of all the coordinates of all the points, we can consider these expressions as fixed Boolean functions of the binary representation of the coordinates of robots and shelters, i. e. just as the well-known Boolean function. We can apply the split computation protocol to this function [17].

Conclusion. Basic concepts of agents and multi-agent systems have been discussed in the article. A review of multi-agent systems has been performed. Such main characteristics of the agent have been provided as autonomy, proactivity, social ability, and reactivity; also, agents can possess such additional characteristics as resilience, reasonability, productivity, mobility, personality, and rationality. A number of new multi-agent algorithms have been presented, and their correctness has been proved. Two propositions for solving RAM and MRP problems have been proved.

А. Сатыбалдиева[1]**, А. Исмаилова**[1]**,**
Р. Молдашева[1]**, А. Муханова**[2]**, Қ. Қадірқұлов**[1]

[1]С. Сейфуллин атындағы Қазақ агротехникалық университеті, Нұр-Сұлтан қаласы, Қазақстан;
[2]Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Нұр-Сұлтан қаласы, Қазақстан

БІЛІМДЕРДІ БЕЙНЕЛЕУГЕ АРНАЛҒАН ДЕРЕКТЕРДІҢ АБСТАРКТЫ ТИПІ ЖӘНЕ КӨП АГЕНТТІК ЖҮЙЕЛЕРДІҢ СПЕЦИФИКАЦИЯЛАРЫ

Аннотация. Көп агенттік алгоритмдермен шешілетін көптеген мәселелерді әлеуметтік бағдарламалық жасақтама (Social Software) проблемаларының мысалдары деп санауға болады - бұл салыстырмалы түрде жаңа ғылыми парадигма, оның мәні келесіде. Қазіргі әлемде көптеген әлеуметтік талаптар мен процедуралар өте айқын сипатталған алгоритмдік сипатқа ие. Сондықтан бұл талаптарды (жартылай) формальды спецификациялар түрінде, ал процедураларды бағдарламалық түрде (кейбір бағдарламалау тілінде немесе жартылай формалы псевдокодта) ұсынуға болады, бұл процедуралардың қасиеттерін бағдарламалық талдау және тексеру әдістерімен зерттеуге болады.

Мақалада агенттер мен көп агенттік жүйелер туралы негізгі түсініктер қарастырылады. Сонымен қатар, әлеуметтік-бағдарламалық жасақтама контекстінде білімді ұсынудың көп агенттік алгоритмдерінің екі мәселесі қарастырылған. Бірқатар жаңа агенттер алгоритмдері ұсынылды және олардың дұрыстығы дәлелденді. Агенттің автономия, белсенділік, әлеуметтік қабілеттілік, реактивтілік сияқты негізгі сипаттамалары келтірілген, сонымен қатар агенттер табандылық, парасаттылық, өнімділік, ұтқырлық, жеке тұлға, парасаттылық сияқты қосымша сипаттамаларға ие бола алады. Бірқатар жаңа агенттер алгоритмдері ұсынылды және олардың дұрыстығы дәлелденді. RAM және MRP мәселелерін шешуге арналған екі тұжырым дәлелденді.

Түйін сөздер: мультиагенттік жүйелер, мультиагенттік алгоритмдер, әлеуметтік-бағдарламалық инженерия, RAM және MRP тапсырмалары, формальды сипаттамалар, үлестірілген жүйе, SWP алгоритмі, верификация.

А. Сатыбалдиева[1], А. Исмаилова[1], Р. Молдашева[1], А. Муханова, К. Кадиркулов[1]

[1]Казахский агротехнический университет им. С.Сейфуллина, Нур-Султан, Казахстан;
[2]Евразийский национальный университет им.Л.Н.Гумилева, Нур-Султан, Казахстан

АБСТРАКТНЫЕ ТИПЫ ДАННЫХ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ И СПЕЦИФИКАЦИИ МУЛЬТИАГЕНТНЫХ СИСТЕМ

Аннотация. Многие задачи, которые решаются мультиагентными алгоритмами можно считать примерами задач социо-программной инженерией (Social Software) – сравнительно новой научной парадигмы, суть которой состоит в следующем. В современном мире очень многие социальные требования и процедуры носят очень чётко описанный алгоритмический характер. Поэтому эти требования можно представить в виде (полу) формальных спецификаций, а процедуры – программно (на каком-либо языке программирования или на полуформальном псевдокоде), после чего свойства этих процедур можно исследовать методами анализа и верификации программ.

В статье рассматриваются основные понятия агентов и мультиагентных систем. Также рассматриваются две проблемы мультиагентных алгоритмов для представления знаний в контексте социо-программной инженерии. Представлен ряд новых мультиагентных алгоритмов, доказана их корректность. Приведены основные характеристики агента, такие как автономность, проактивность, социальная способность, реактивность, также агенты могут иметь дополнительные характеристики, такие как стойкость, рассуждаемость, производительность, мобильность, личность, рациональность. Представлены ряд новых мультиагентных алгоритмов, доказана их корректность. Доказаны два утверждения для решения RAM и MRP задач.

Ключевые слова: мультиагентные системы, мультиагентные алгоритмы, социо-программная инженерия, RAM и MRP задачи, формальные спецификаций, распределённая система, алгоритм SWP, верификация.

Information about authors:

Satybaldiyeva A., PhD, Senior Lecturer of Department “Information systems”, S. Seifullin Kazakh Agrotechnical University, satekbayeva@gmail.com, https://orcid.org/0000-0001-5740-7934;

Ismailova A., PhD, Senior Lecturer of Department “Information systems”, S. Seifullin Kazakh Agrotechnical University, a.ismailova@mail.ru, https://orcid.org/0000-0002-8958-1846;

Moldasheva R., PhD Student of Department “Information systems”, S. Seifullin Kazakh Agrotechnical University, raushan85_07@mail.ru, https://orcid.org/0000-0002-4570-0487;

Mukhanova Ayagoz Asanbekovna – Ph.D., Associate Professor of the Department “Information systems”, L.N. Gumilyov Eurasian National University, Satbayev str, 2, Nur-Sultan, 010000, Kazakhstan.
E-mail: ayagoz198302@mail.ru, https://orcid.org/0000-0003-3987-0938;

Kadirkulov K., PhD Student of Department “Information systems”, S. Seifullin Kazakh Agrotechnical University, kkuanysh@gmail.com, https://orcid.org/0000-0003-0506-4890

REFERENCES

[1] Weiss G. Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence. Cambridge: Press, 1999. P. 79-120.

[2] Stone P., Veloso M. Multi-agent Systems: A Survey from a Machine Learning Perspective // Journal of Artificial Intelligent Research. 2007. Vol. 24, №6. P. 547-578.

[3] Weizenbaum J. Computer Power and Human Reason. San Francisco, 1976. 542 p.

[4] Seipp A. Intelligent Agent // School of Library, Archival and Information Studies. – Columbia: University of British Columbia, 2001. 140 p.

[5] Coen M. SodaBot: A Software Agent Environment and Construction System: MIT AI Lab Technical Report, 1994. P. 1493-1500 // http://citeseer.ist.psu.edu/coen94sodabot.html.

[6] Maes P. Artificial Life Meets Entertainment: Lifelike Autonomous Agents // Communications of the ACM. 1995. Vol. 38, №11. P. 108-114.

[7] Hayes-Roth B. An Architecture for Adaptive Intelligent Systems // Artificial Intelligence: Special Issue on Agent Interactivity. 1995. Vol. 72, № 1-2. P. 329-365.

[8] Wooldridge M. An Introduction to Multiagent Systems. Kyoto, 2002. 255 p.

[9] Honavar V. Intelligent Agents and Multi-agent Systems. John Willey & Sons Ltd, 1998. 302 p. // http://www.cs.iastate.edu/~honavar.

[10] Sycara K.P. The Many Faces of Agents. AI Magazine. California, USA, 1998 // http://www.aaai.org/AITopics/assets/PDF/AIMag19-02-1-article.pdf.

[11]. A. Yu., Bernshtein, N. V. Shilov. Multi-agent problem about robots in space: complexity, information, and cryptographic aspects // Modeling and analysis of information systems. 2012. V. 20, No. 2. Pages 34-53.

[12] Bodin E.V., Garanina N.O., Shilov N.V. Mars Robot Puzzle (a Multiagent Approach to the Dijkstra Problem) // Modeling and analysis of information systems. 2001. №18(2). P. 111-126.

[13] Brams S.J., Taylor A.D. Fair Division – From cake-cutting to dispute resolution. Cambridge: Cambridge University Press, 1996. 324 p.

[14] Fagin R., Halpern J.Y., Moses Y., Vardi M.Y. Reasoning about Knowledge. London: MIT Press, 1995. 345 p.

[15] Goldreich O. Foundations of Cryptography - A Primer // Foundations and Trends in Theoretical Computer Science. 2005. №1. P. 1-116.

[16] Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code C. New York: John Willey&Sons, 1996. 276 p.

[17] Some Results on Multiagent Algorithms in Social Computing/Software Context. International Journal of INFORMATION (Indexed by Scopus, JDream, Mathematical Reviews, Zentralblatt MATH, ProQuest, Swets, EBSCO). 2014. Vol.17, No.1. P. 229-240.